Engineering Tripos Part IIB FOURTH YEAR

Module 4F12: Computer Vision and Robotics

Solutions to 2005 Tripos Paper

1. **Feature detection**

(a) Edge detection is commonly used in the first stage of many computer vision applications, since edges provide a compact representation of image structure and are invariant to illumination effects. Compared with raw images, edges offer significant data reduction while preserving much of the image's useful information content (it is possible to recognise many structures in a line drawing of a scene). In contrast, most of the discarded information is not useful for discovering scene structure and motion.

(b) In edge detection the smoothed pixels $s(x)$ are obtained by discrete convolution with a Gaussian kernel. Intensity discontinuities are localised by differentiating the smoothed pixels to obtain the gradient $d(x)$ and looking for the local maxima.

(c) The rate of change of intensity I in the direction $\mathbf{n}$ is found by taking the scalar product of ∇I and $\hat{\mathbf{n}}$:

$$I_n \equiv \nabla I(x,y).\hat{\mathbf{n}} \Rightarrow I_n^2 = \frac{\mathbf{n}^T \nabla I \nabla I^T \mathbf{n}}{\mathbf{n}^T\mathbf{n}} = \frac{\mathbf{n}^T \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \mathbf{n}}{\mathbf{n}^T\mathbf{n}}$$

where $I_x \equiv \partial I/\partial x$, etc.

We smooth I_n^2 by convolution with a Gaussian kernel:

$$C_n(x,y) = G_\sigma(x,y) * I_n^2 = \frac{\mathbf{n}^T \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \mathbf{n}}{\mathbf{n}^T\mathbf{n}}$$

where $\langle\ \rangle$ is the smoothed value. The smoothed change in intensity in direction $\mathbf{n}$ is therefore given by

$$C_n(x,y) = \frac{\mathbf{n}^T \mathrm{A} \mathbf{n}}{\mathbf{n}^T\mathbf{n}}$$

where A is the 2×2 matrix

$$\begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix}$$

Elementary eigenvector theory tells us that

$$\lambda_1 \leq C_n(x,y) \leq \lambda_2$$

where λ_1 and λ_2 are the eigenvalues of A. So, if we try every possible orientation $\mathbf{n}$, the maximum change in intensity we will find is λ_2, and the minimum value is λ_1.

We can detect a corner by looking at the eigenvectors of A. For a (corner) λ_1 and λ_2 both large. It is necessary to calculate A at every pixel and mark corners where the quantity $\lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$ exceeds some threshold ($\kappa \approx 0.04$ makes the detector a little "edge-phobic"). Note that $\det \mathrm{A} = \lambda_1\lambda_2$ and $\mathrm{trace\ A} = \lambda_1 + \lambda_2$, so the required eigenvalue properties can be obtained directly from the elements of A.

2. **Camera calibration and vanishing points**

(a) The relationship is valid under the assumption that the image is formed by a "pinhole camera", such that rays pass through a single point (the optical centre) before striking the image plane. The relationship does not account for nonlinear distortion, which affects all real cameras to some extent.

(b)The projection matrix can be written in the form

$$\mathrm{P} = [\mathrm{K}][\mathrm{R}|\mathbf{T}]$$

where R is the rotation matrix between camera and world coordinates, $\mathbf{T}$ is the translation vector between camera and world coordinates, and K is the 3×3 matrix of the camera's intrinsic parameters:

$$\mathrm{K} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

(In more detail) We need to decompose the left 3×3 sub-matrix of P into an upper triangular matrix K and an orthogonal (rotation) matrix R. This can be achieved using QR decomposition. It is not possible to decouple the focal length from the pixel scaling factors. Given the projective camera matrix, we can attempt to recover the intrinsic and extrinsic parameters using QR decomposition. Writing

$$\begin{aligned} \mathrm{P} &= \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[\begin{array}{ccc|c} & \mathrm{R} & & \mathbf{T} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{c|c} \mathrm{R} & \mathbf{T} \end{array}\right] \\ &= \mathrm{K}\left[\begin{array}{c|c} \mathrm{R} & \mathbf{T} \end{array}\right] = \left[\begin{array}{c|c} \mathrm{KR} & \mathrm{K}\mathbf{T} \end{array}\right] \end{aligned}$$

(c) Distant points on lines parallel to the world X-axis can be represented in homogeneous coordinates as

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Applying the projection matrix P, we find the image of these points is

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \end{bmatrix}$$

The vanishing point of lines which are parallel to the world X-axis is therefore $(u, v) = (p_{11}/p_{31}, p_{21}/p_{31})$.

(d) P can be estimated by observing the images of known 3D points. Each point we observe gives us a pair of equations:

$$u = \frac{su}{s} = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

$$v = \frac{sv}{s} = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}$$

Since we are observing a known scene, we know X, Y, and Z, and we observe the pixel coordinates u and v in the image. So we have two linear equations in the unknown camera parameters. Since there are 11 unknowns (the overall scale of P does not matter), we need to observe at least 6 points to calibrate the camera.

The equations can be solved using orthogonal least squares. First, we write the equations in matrix form:

$$\mathbf{Ap} = \mathbf{0}$$

where $\mathbf{p}$ is the 12×1 vector of unknowns (the twelve elements of P), A is the $2n \times 12$ matrix of coefficients and n is the number of observed calibration points. The orthogonal least squares solution corresponds to the eigenvector of $\mathrm{A}^T\mathrm{A}$ with the smallest corresponding eigenvalue.

The linear solution is, however, only approximate, since we have not taken into account the special structure of P. Ideally, the linear solution should be used as the starting point for nonlinear optimization, finding the parameters of the rigid body transformation, perspective projection and CCD mapping that minimize the errors between measured image points (u_i, v_i) and projected (or modelled) image positions $(\hat{u}_i, \hat{v}_i)$:

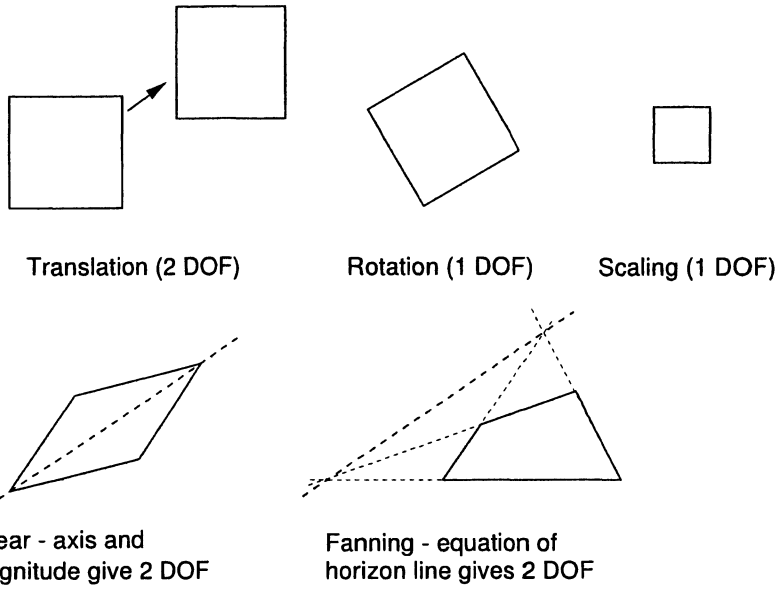
$$\min_{\mathbf{P}} \sum_i ((u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2)$$

It is essential that the calibration points are not coplanar, since otherwise we are not exercising all the degrees of freedom of the camera model and the set of linear equations will not be independent. Consequently, the least squares procedure will not find a unique solution (there will be a degenerate zero eigenvalue).

3. **Projective Transformations**

(a) Since the transformation operates on homogeneous coordinates, the overall scale of the transformation matrix does not matter and we could, for instance, set t_{33} to 1. The transformation therefore has 8 degrees of freedom.

The image of a square could take any of the following forms:



(b) Before the camera is rotated, assume, without loss of generality, that the camera is aligned with the world coordinate system and hence

$$\tilde{\mathbf{w}} = \mathrm{C}\left[\ \mathrm{I}\ |\ \mathbf{O}\ \right]\tilde{\mathbf{X}} = \mathrm{C}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathrm{C}\mathbf{X}$$

It follows that

$$\mathbf{X} = \mathrm{C}^{-1}\tilde{\mathbf{w}}$$

After rotating by R about the optical centre, the same world point $\mathbf{X}$ projects to a different image point $\tilde{\mathbf{w}}'$ as follows:

$$\tilde{\mathbf{w}}' = \mathrm{C}\left[\ \mathrm{R}\ |\ \mathbf{O}\ \right]\tilde{\mathbf{X}} = \mathrm{CR}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathrm{CR}\mathbf{X} = \mathrm{CRC}^{-1}\tilde{\mathbf{w}}$$

Hence the relationship between points in the original image and corresponding points in the second image is a plane to plane projectivity. The projectivity can be estimated by observing at least four corresponding points in the two images. Each correspondence gives a constraint of the form

$$\begin{bmatrix} su' \\ sv' \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

By rearranging this matrix equation, it becomes clear how each correspondence provides two linear equations in the unknown elements of P:

$$u' = \frac{su'}{s} = \frac{p_{11}u + p_{12}v + p_{13}}{p_{31}u + p_{32}v + p_{33}}$$

$$v' = \frac{sv'}{s} = \frac{p_{21}u + p_{22}v + p_{23}}{p_{31}u + p_{32}v + p_{33}}$$

The set of constraints can be written in matrix form:

$$\mathbf{Ap} = \mathbf{0}$$

where $\mathbf{p}$ is the 9×1 vector of unknowns (the 9 elements of P), A is the $2n \times 9$ matrix of coefficients and n is the number of corresponding points observed in the two images. This can be solved using orthogonal least squares.

The mosaic can be constructed as follows. The camera is rotated around the optical centre and a sequence of images is acquired, with each image overlapping its predecessor to some extent (say 50%). The plane to plane projectivity P relating consecutive pairs of images is estimated using correspondences in the overlap region. The correspondences can be located manually, or perhaps even automatically using some sort of correlation scheme. P is then used to *warp* one image into the coordinate frame of its predecessor, by finding the grey level $I(\tilde{\mathbf{w}})$ in the second image associated with each pixel $\tilde{\mathbf{w}}'$ in the frame of the first image. The two images can then be displayed in the same frame. Some sort of blending is required in the overlap region. This process is repeated for all pairs of images, allowing the entire sequence to be displayed in a single frame. If all has gone well (and the camera has not been translated as well as rotated), the seams should be invisible in the final composite mosaic.

4. **Stereo vision**

The fundamental matrix F relates points in the left and right images of a stereo pair:

$$\tilde{\mathbf{w}}'^T \mathrm{F} \tilde{\mathbf{w}} = 0$$

where $\tilde{\mathbf{w}} = (u, v, 1)$ are the point's pixel coordinates in the left image, and $\tilde{\mathbf{w}}'$ are the coordinates of the corresponding point in the right image. The constraint arises from the requirement that the rays from the two cameras' optical centres through $\tilde{\mathbf{w}}$ and $\tilde{\mathbf{w}}'$ must intersect at a point in space. F has zero determinant and can be determined only up to scale.

F can be estimated from point correspondences. Each point correspondence $\tilde{\mathbf{w}} \leftrightarrow \tilde{\mathbf{w}}'$ generates one constraint on F:

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$

This is a linear equation in the unknown elements of F. Given eight or more perfect correspondences (image points in *general* position, no noise), F can be determined uniquely up to scale by solving the simultaneous linear equations. In practice, there may be more than eight correspondences and the image measurements will be noisy. The system of equations can then be solved by least squares, or using a robust regression scheme to reject outliers.

The linear technique does not enforce the constraint that $\det \mathrm{F} = 0$. If the eight image points are noisy, then the linear estimate of F will *not* necessarily have zero

determinant and the epipolar lines will not meet at a point. Nonlinear techniques exist to estimate F from 7 point correspondences, enforcing the rank 2 constraint.

The corresponding result for the right epipole is $\mathrm{F}^T\tilde{\mathbf{w}}'_{\mathrm{e}} = \mathbf{0}$.

Roberto Cipolla
February 2005